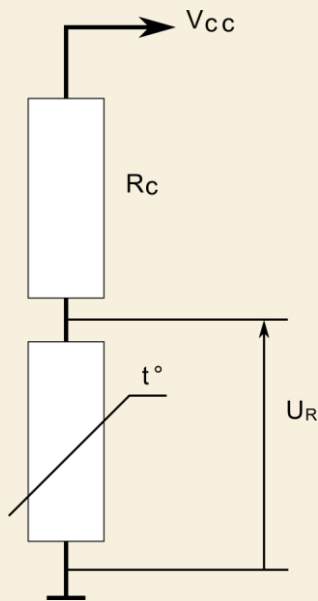


# **СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ**

СТЗ. ОСНОВНЫЕ  
ПОНЯТИЯ.  
ЦВЕТОВЫЕ  
ПРОСТРАНСТВА.

# ВТОРАЯ ЗАДАЧА – ПОПРОБУЕМ ПОСТРОИТЬ КОРРЕКТНУЮ МОДЕЛЬ



- Показания индикатора

$$Z = 1024 \frac{U_R}{U_{оп}} = 1024 \cdot \frac{R}{R_c + R}$$
$$\approx 1024 \cdot \frac{R}{R_c + R_0}$$

- Зависимость R - экспоненциальная

$$R = R_0 \cdot 10^{K(T-T_0)}$$

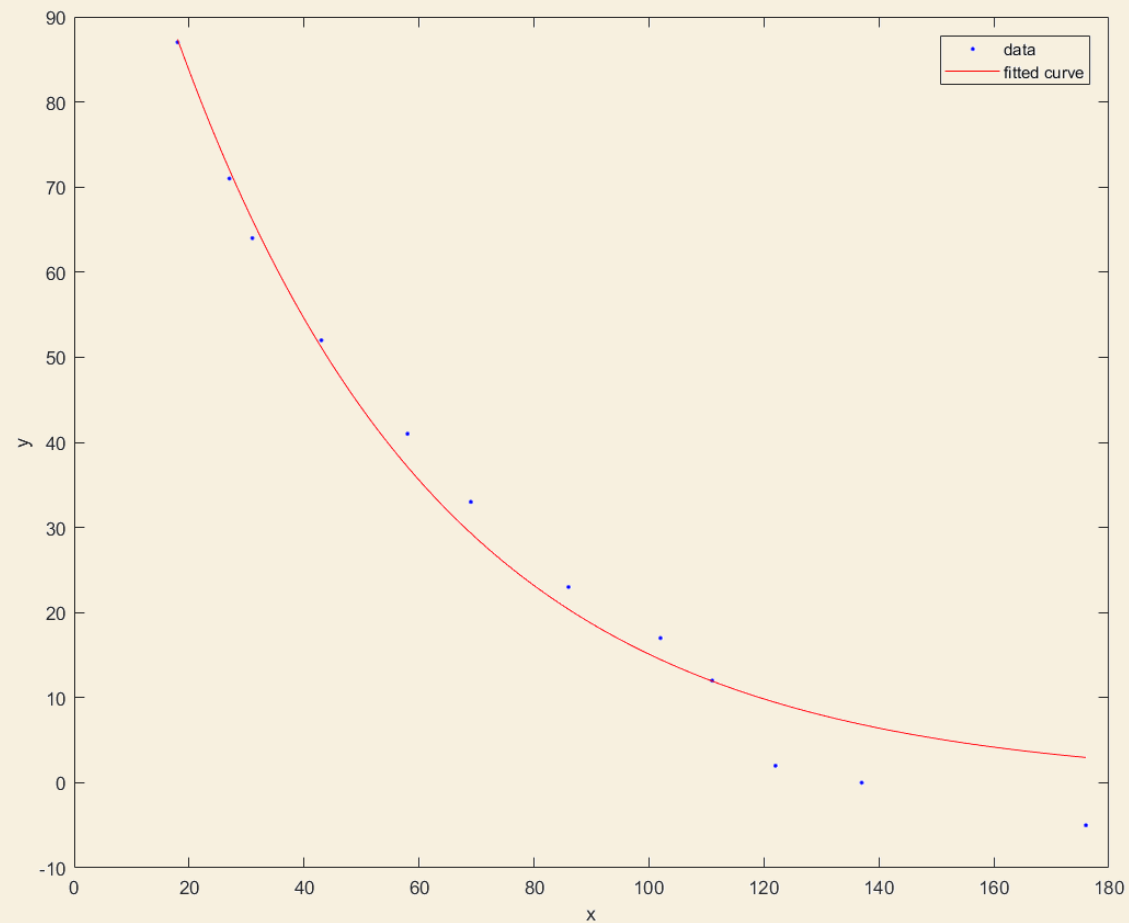
Тогда

$$Z \approx 1024 \cdot \frac{R_0 \cdot 10^{-T_0 K}}{R_c + R_0} 10^{KT}$$

$$Z \approx C e^{K_1 T}$$

# МОЖНО И БЕЗ «ФИЗИКИ»...

№	Z	T,°C
1	27	71
2	31	64
3	43	52
4	58	41
5	69	33
6	86	23
7	102	17
8	111	12
9	122	2
10	137	0
11	18	87
12	176	-5



# ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЯ НА ЭВМ



0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

# СЕМАНТИЧЕСКАЯ ИНФОРМАЦИЯ





# СЕМАНТИЧЕСКАЯ ИНФОРМАЦИЯ

- вне помещения
- город
- уличное движение
- Пекин, Китай
- Пл. Тяньаньмэнь

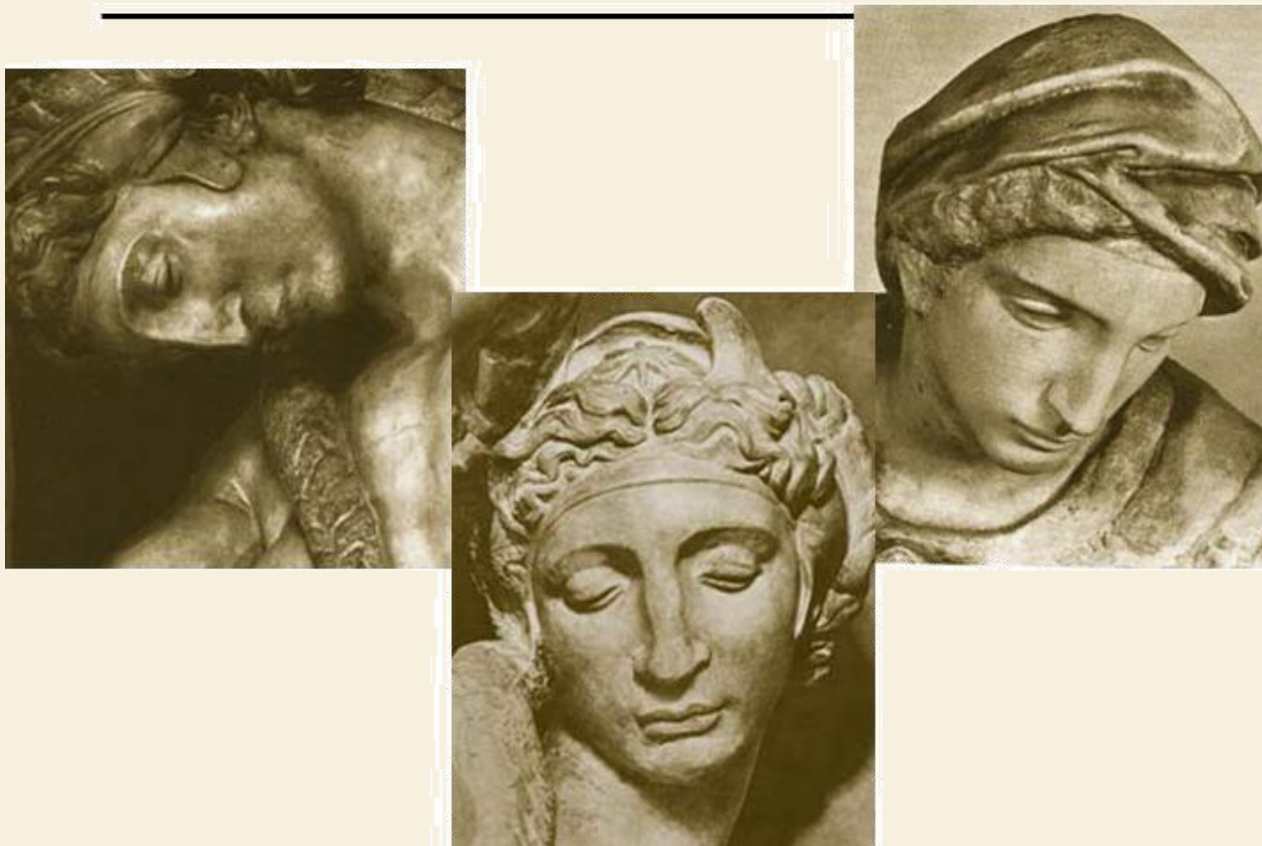


# СЕМАНТИЧЕСКАЯ ИНФОРМАЦИЯ





# ТОЧКА НАБЛЮДЕНИЯ





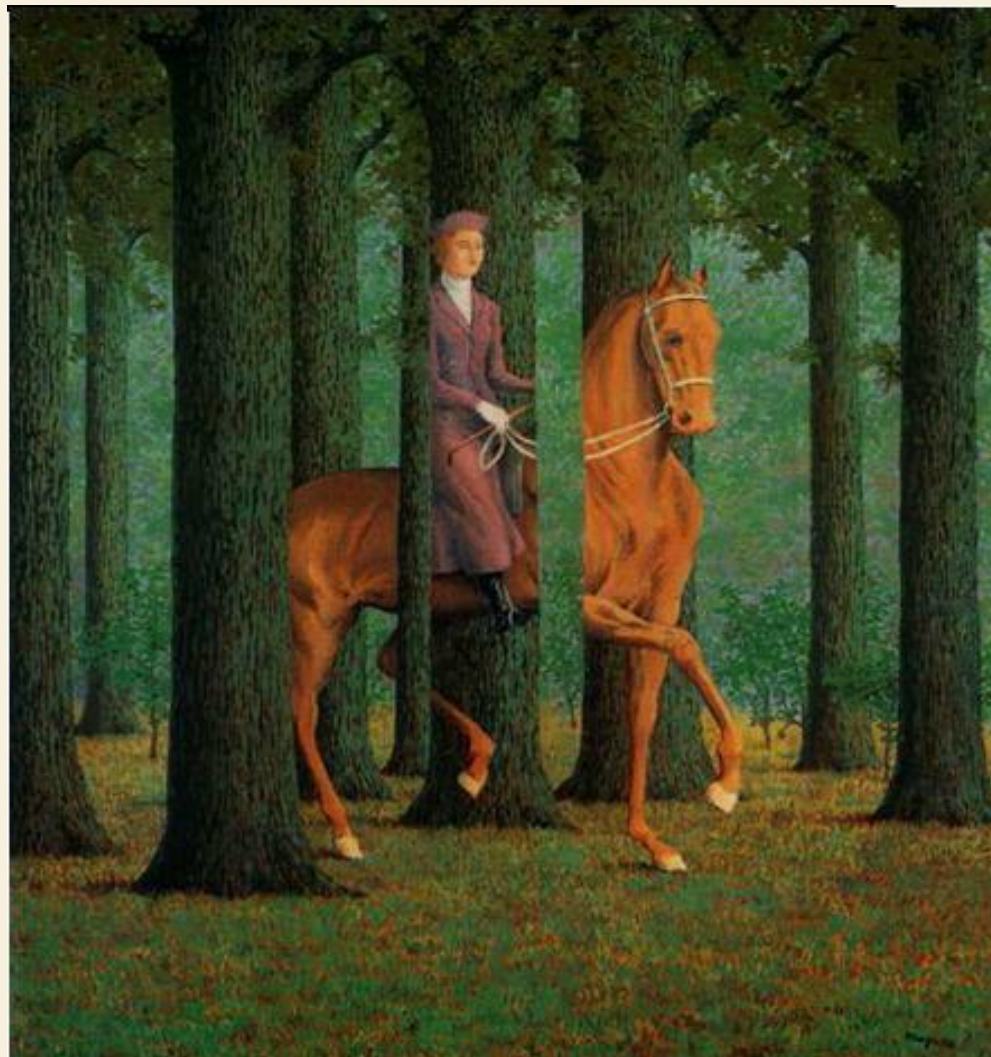
# ОСВЕЩЕНИЕ



# МАСШТАБ



# ПЕРЕКРЫТИЕ



# ДВИЖЕНИЕ





# ВНУТРИКЛАССОВАЯ ИЗМЕНЧИВОСТЬ



# ЗРЕНИЕ – НЕЧЁТКАЯ ЗАДАЧА

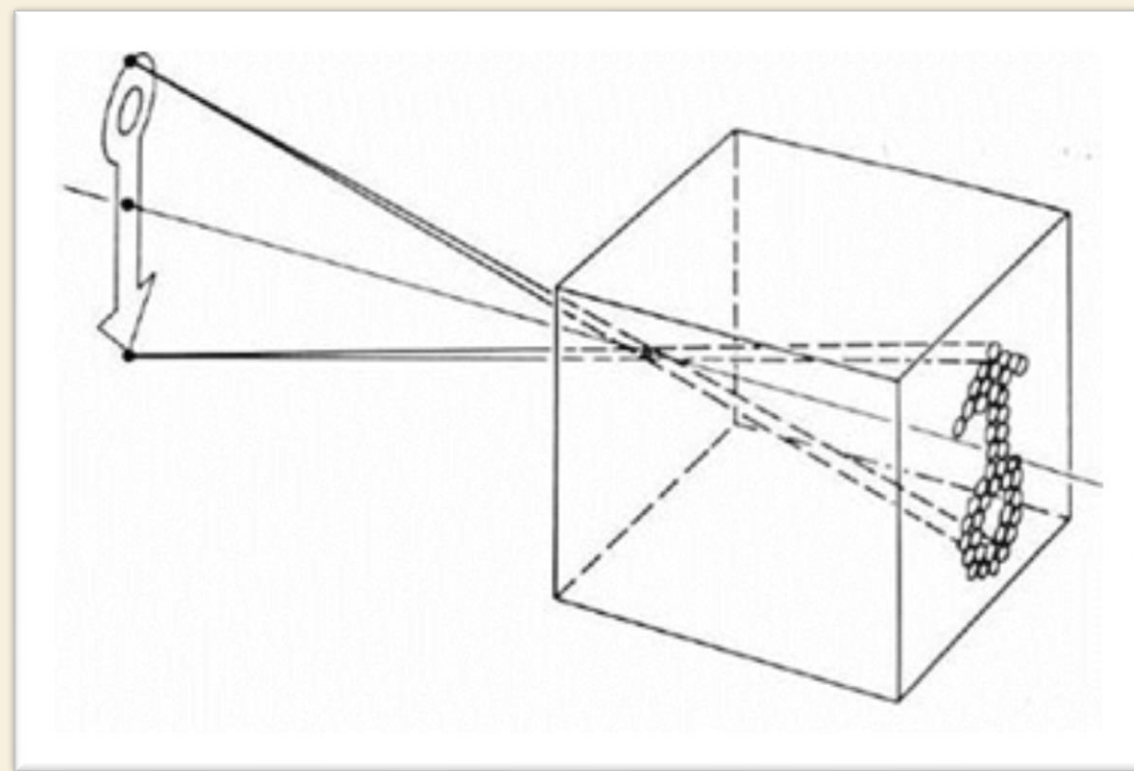
- Разные 3D сцены дают одно и то же 2D изображение
- Необходимы априорные знания о структуре и свойствах мира
- Нам нужно сопоставлять наблюдения и априорные знания



# МОДЕЛЬ КАМЕРЫ-ОБСКУРЫ

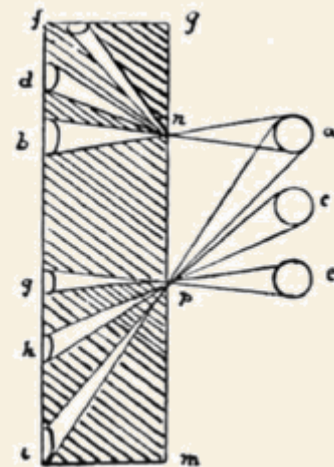
Камера-обскура:

- Захватывает пучок лучей, проходящих через одну точку
- Точечное отверстие называется «Центр проекции» (фокальная точка / **focal point**)
- Изображение формируется на картинной плоскости (**Image plane**)

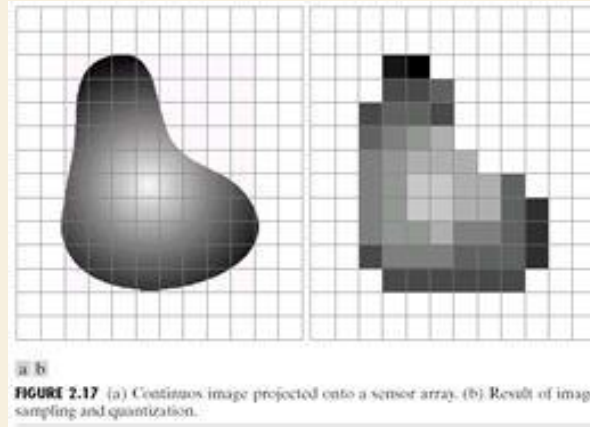
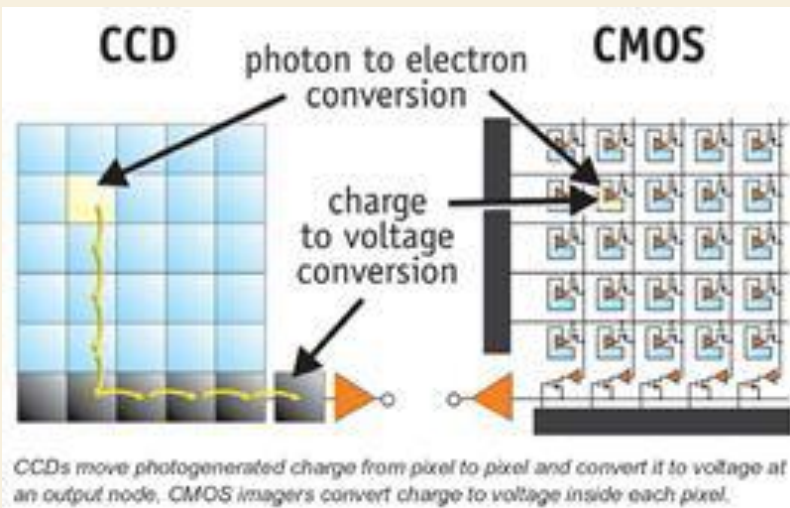


# ПРОБЛЕМЫ

- Теряем углы
- Теряем расстояния и длины
- Перспективные искажения





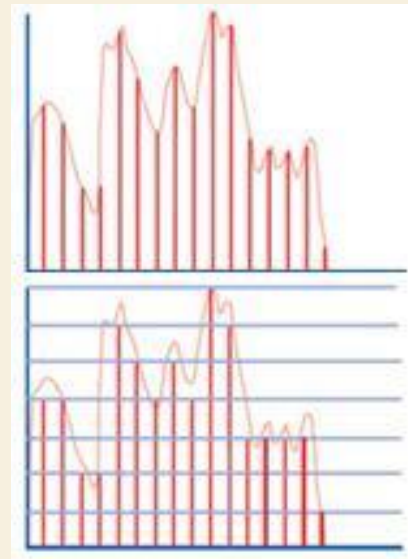
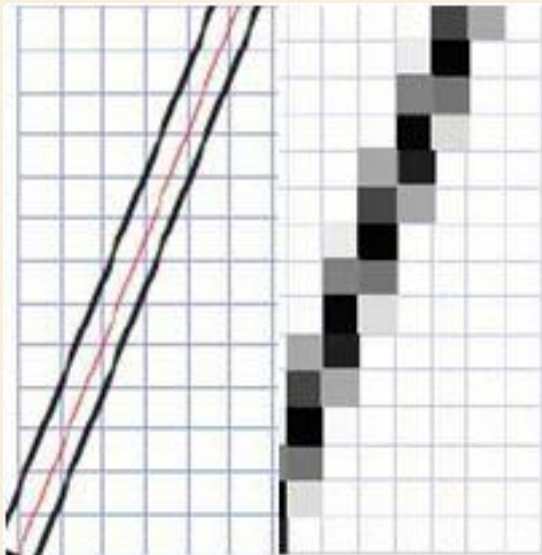


## ЦИФРОВАЯ КАМЕРА - ДИСКРЕТИЗАЦИЯ

Для неподвижного изображения – две дискретизации:

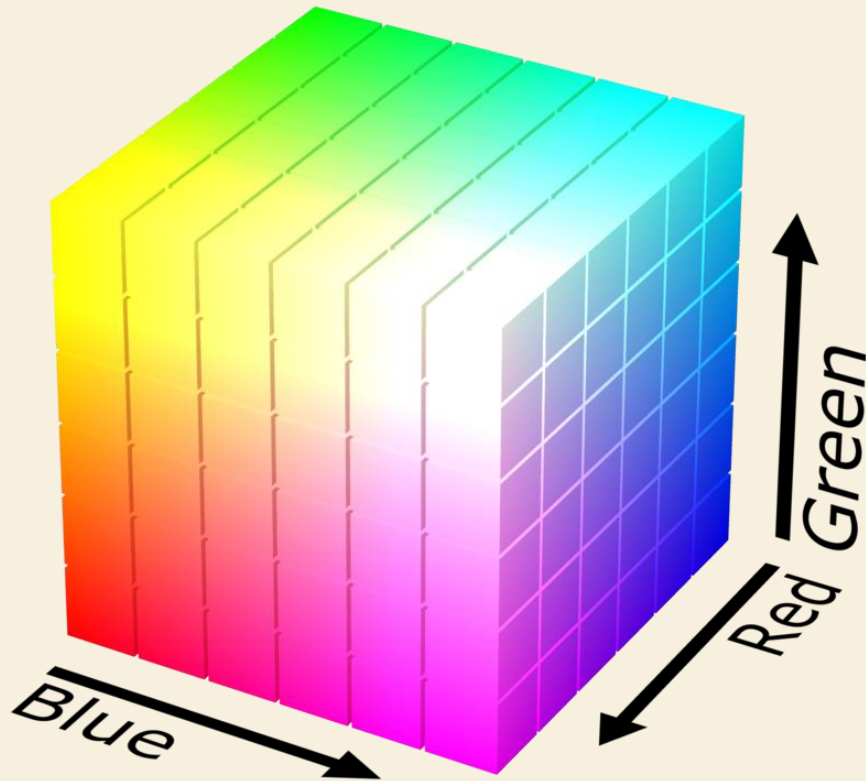
По координате

По уровню сигнала

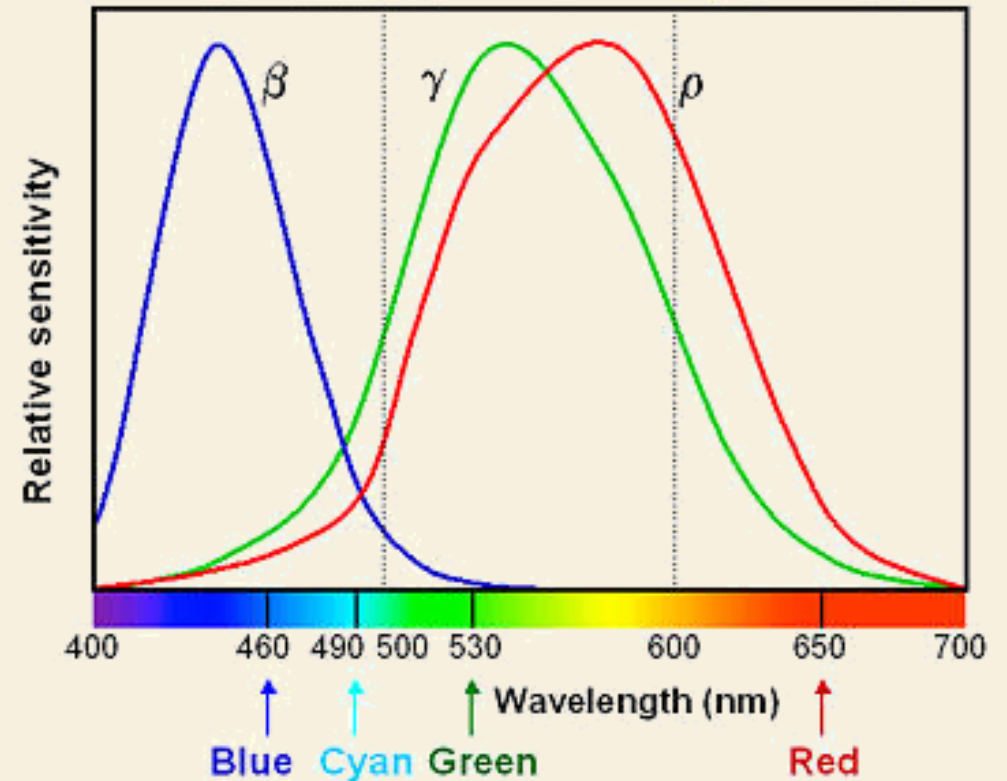


# ЛИНЕЙНЫЕ ЦВЕТОВЫЕ МОДЕЛИ:

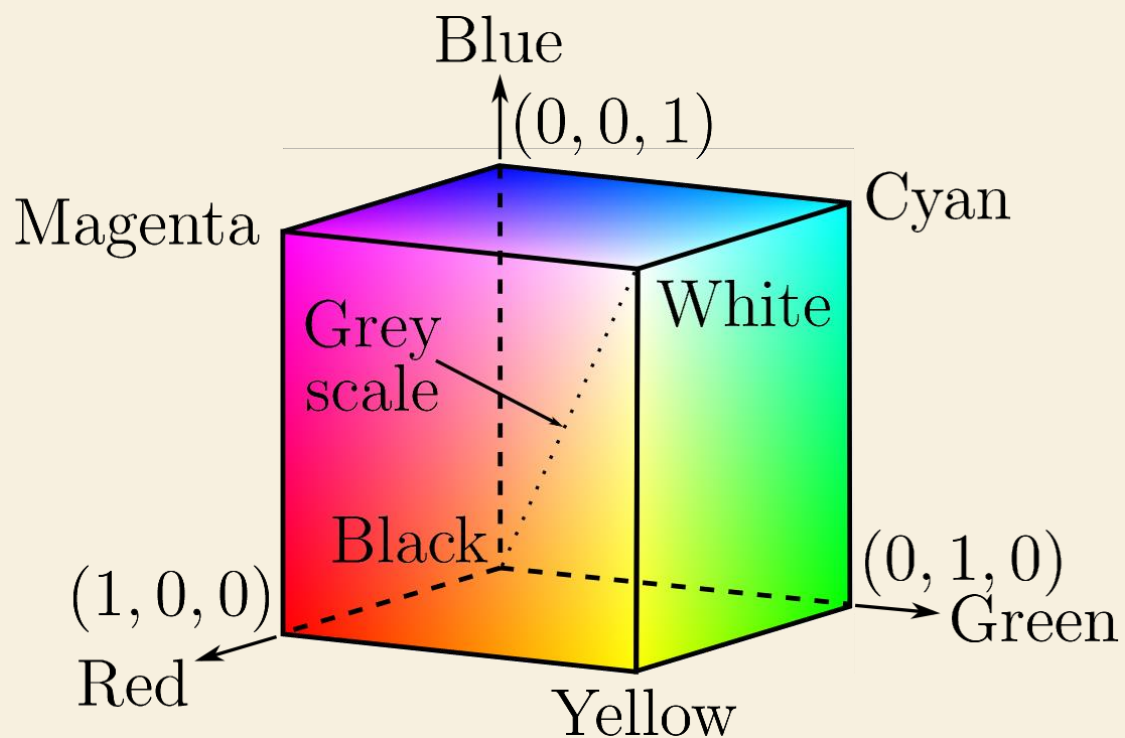
## RGB



Human spectral sensitivity to color  
Three cone types ( $\rho$ ,  $\gamma$ ,  $\beta$ ) correspond *roughly* to R, G, B.



# RGB: ЦВЕТОВОЙ КУБ



- Аддитивная система – RGB
- Субтрактивная система – CMYK

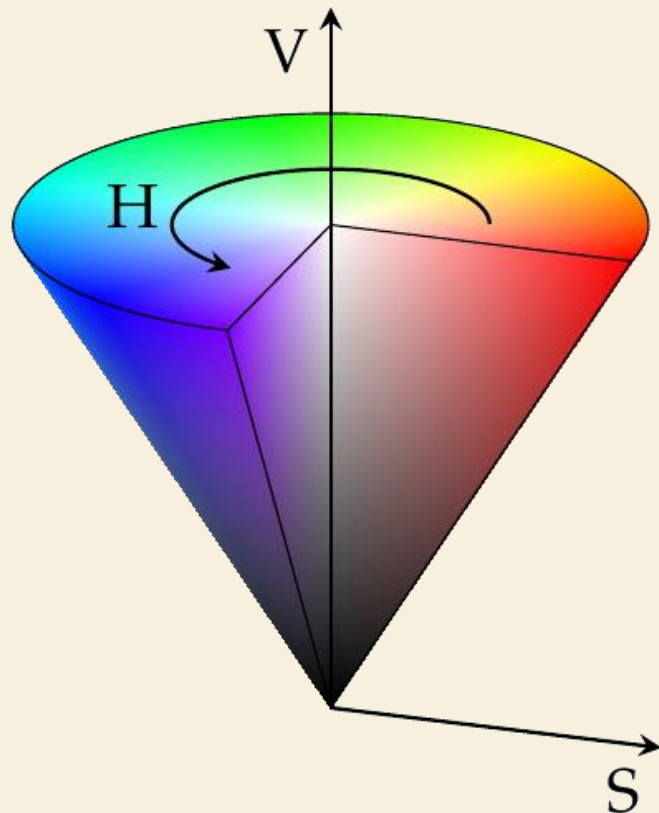
$$C = G + B = W - R$$

$$M = R + B = W - G$$

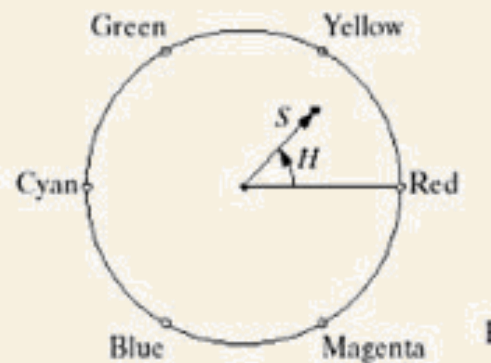
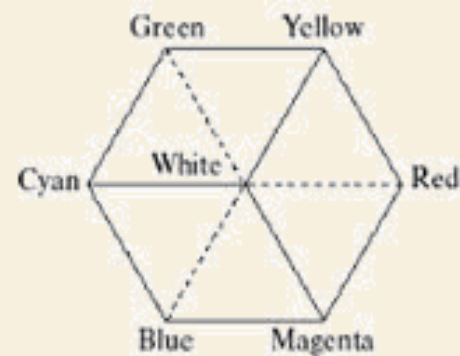
$$Y = R + G = W - B$$

- RGB – на мониторах, CMYK – на печати

# МОДЕЛЬ HSV

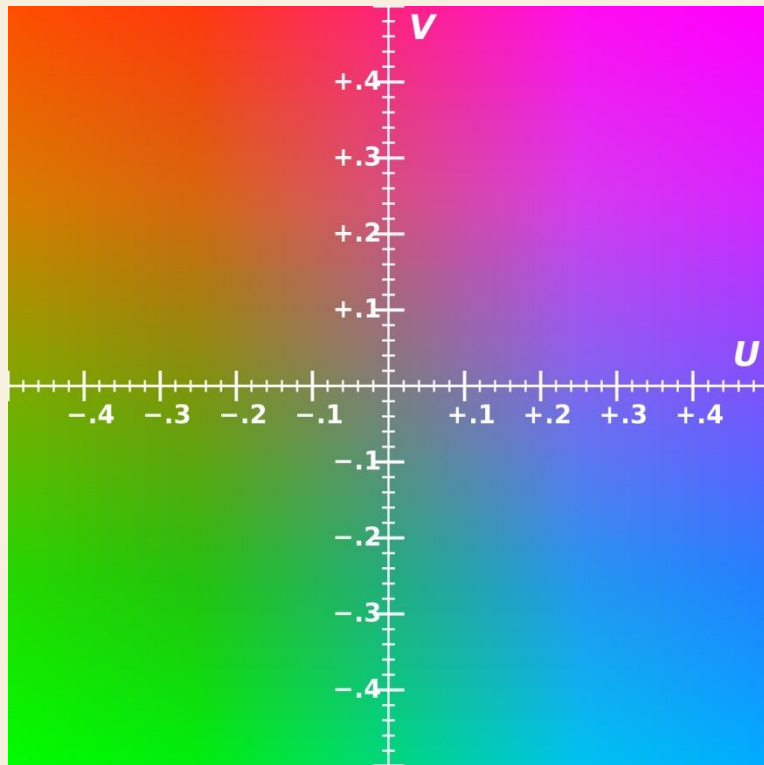


- Hue (Тон)
- Saturation(Насыщенность)
- Value (Интенсивность)



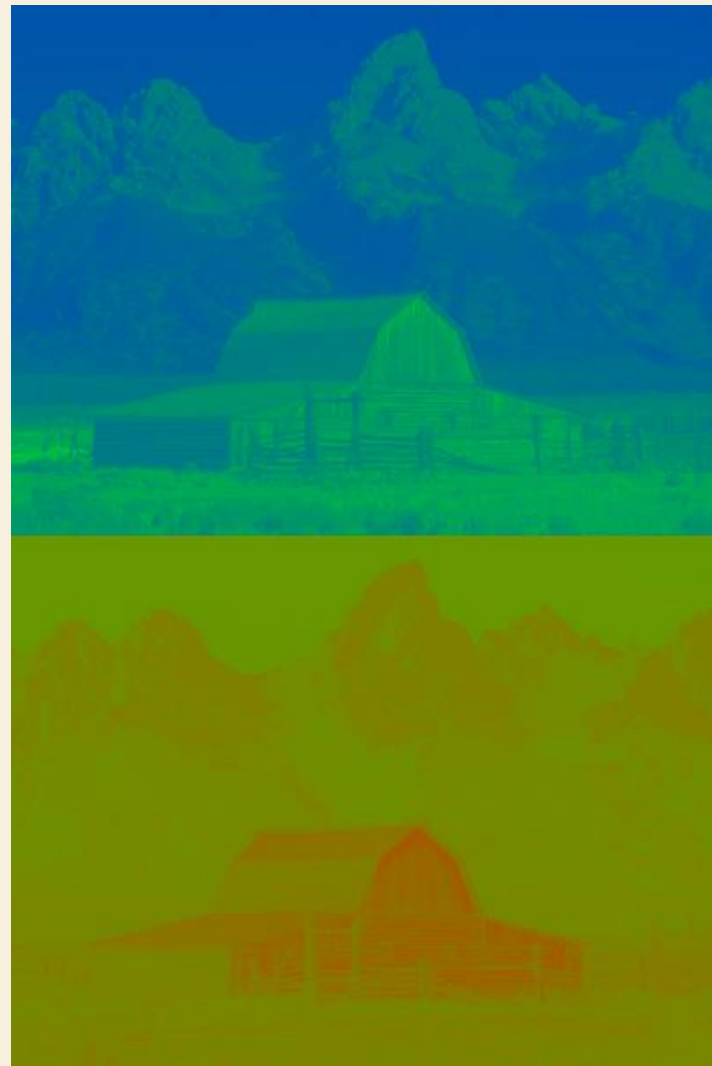
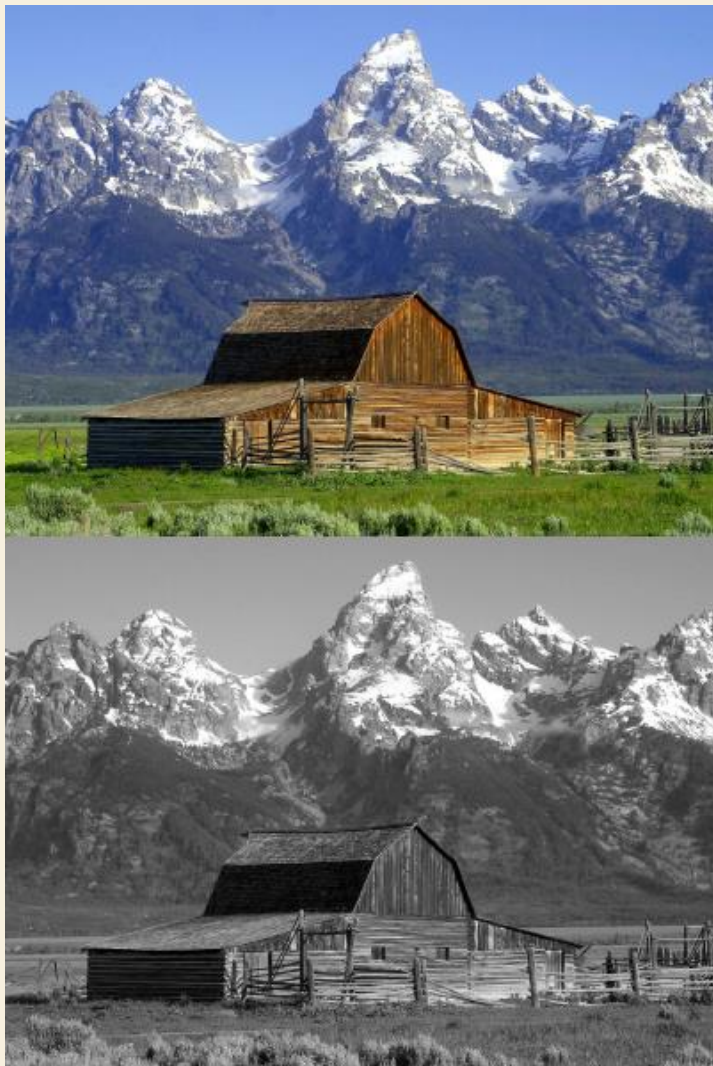


# МОДЕЛЬ YUV...



- $Y$  – яркость (нормированная)
- $U = B - Y$
- $V = R - Y$

# ...И РЕАЛИЗАЦИЯ УСВСР



```

#include "opencv2/highgui.hpp"
#include "opencv2/imgcodecs.hpp"
#include "opencv2/imgproc.hpp"
#include <iostream>
using namespace std;
using namespace cv;
int main(int argc, char** argv)
{
    Mat src, dst;
    String imageName( "Lenna.png" ); // by default
    if (argc > 1)
    {
        imageName = argv[1];
    }
    src = imread( imageName, IMREAD_COLOR );
    if( src.empty() )
    { return -1; }
    imshow("Source window", src);
    Mat grayscale;
    cvtColor(src, grayscale, CV_BGR2GRAY);
    imshow("Grayscale", grayscale);
    waitKey(0); // ждём нажатия на любую кнопку
    return 0;
}

```

## ПРИМЕР

IMREAD\_COLOR -

принудительно читать  
изображение как трёхканальное  
BGR

CV\_BGR2GRAY –

преобразование из  
трёхканального цветоного в  
одноканальное в градациях  
серого

```
TEMPLATE = app
CONFIG += console c++11
CONFIG -= app_bundle
CONFIG -= qt

SOURCES += main.cpp
LIBS += \
#     -L/usr/lib/x86_64-linux-gnu \
-lopencv_core \
-lopencv_highgui \
-lopencv_imgproc \
-lopencv_imgcodecs
```

## ПРОЕКТНЫЙ ФАЙЛ

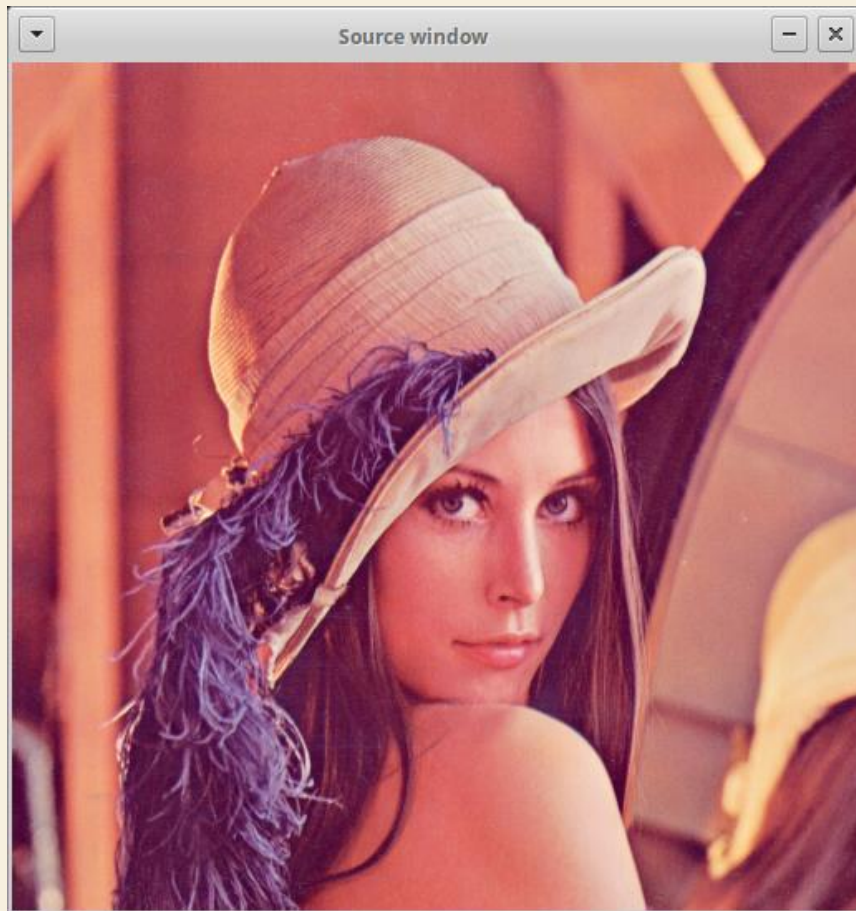
-L<путь> - прописываем ПУТЬ до места расположения библиотек.

В виртуальной машине в этом нет необходимости

-lopencv\_core – подключение libopencv\_core.so и т.д...



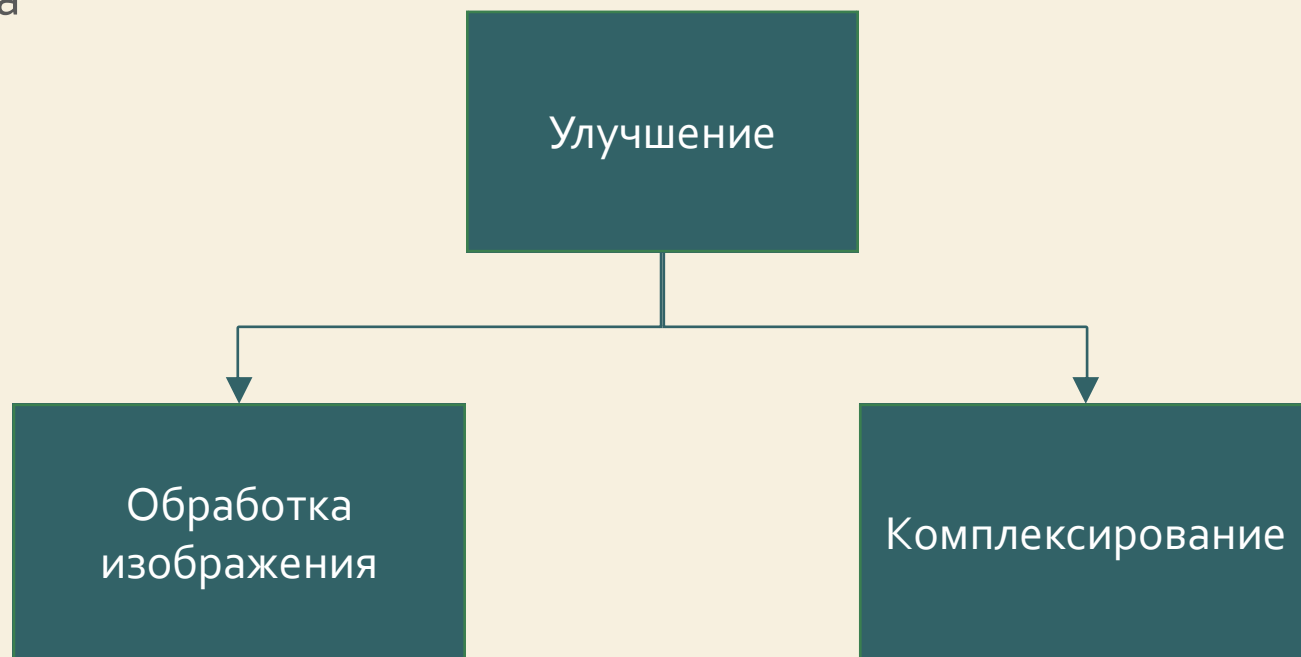
# РЕЗУЛЬТАТ



Используется знаменитое тестовое изображение [Lenna](#)

# УЛУЧШЕНИЕ ИЗОБРАЖЕНИЯ

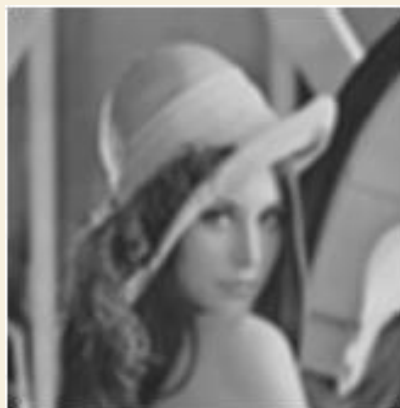
- 2 подхода



# ИЗОБРАЖЕНИЕ: ЧТО МОЖЕТ ПОЛУЧИТЬСЯ ПЛОХО?

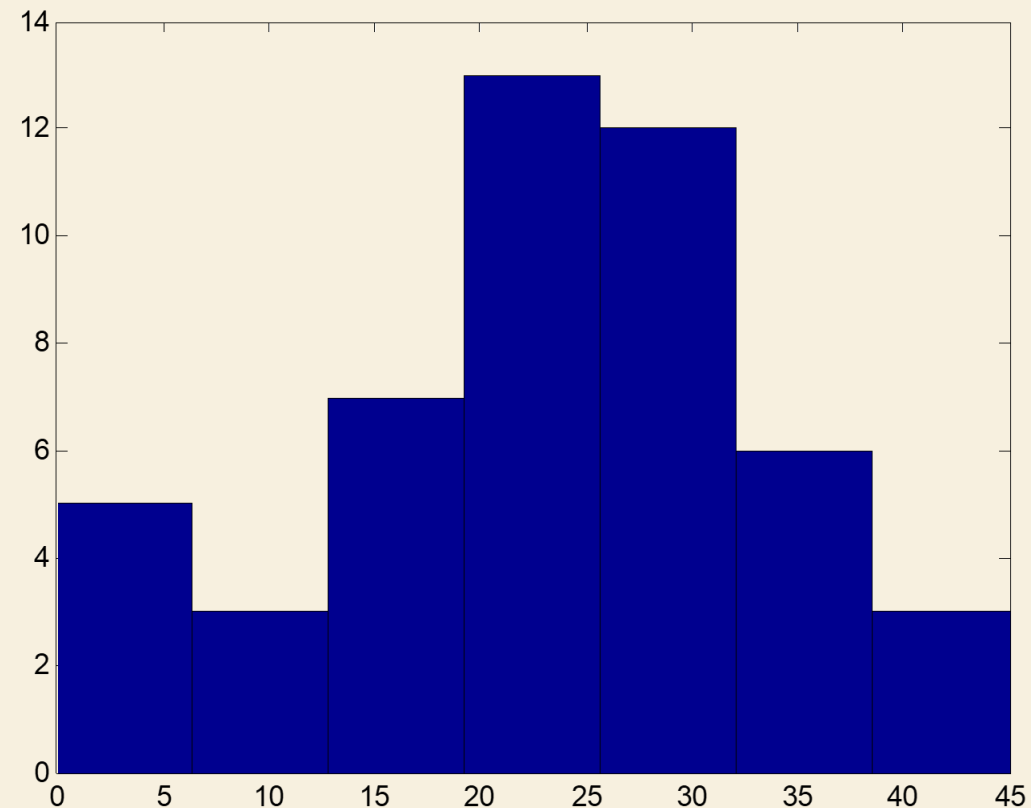


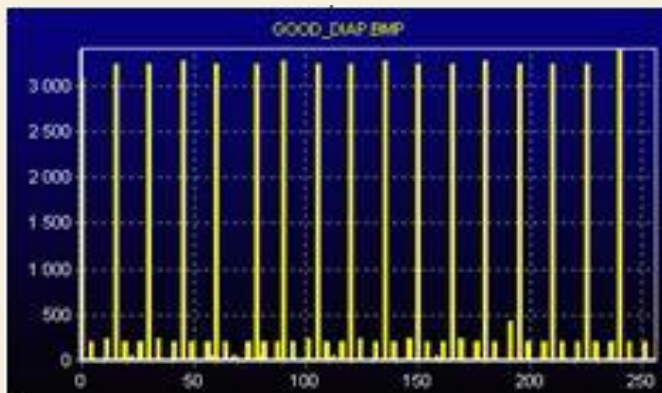
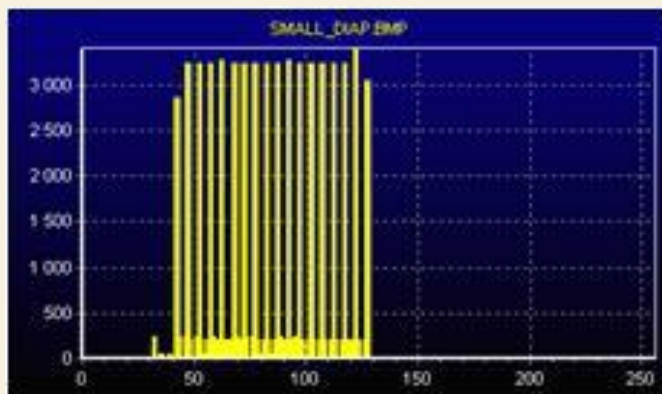
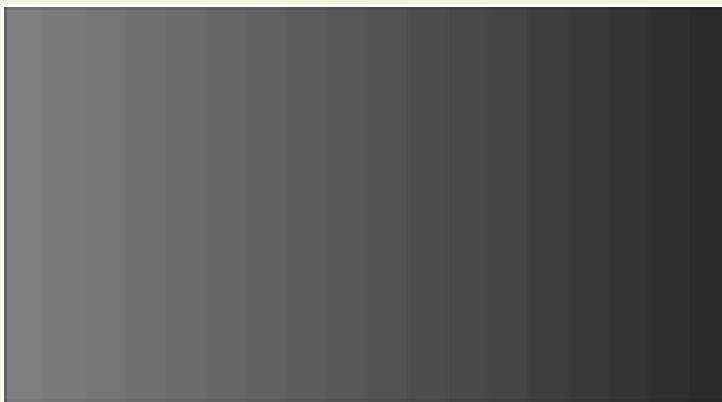
Impulse noise



# ГИСТОГРАММЫ

7	25	26	27	16	32	16
22	27	33	10	25	28	34
39	19	6	23	13	26	6
23	20	17	29	9	35	22
23	43	20	26	32	32	3
15	0	27	33	23	25	33
22	45	6	33	22	16	31





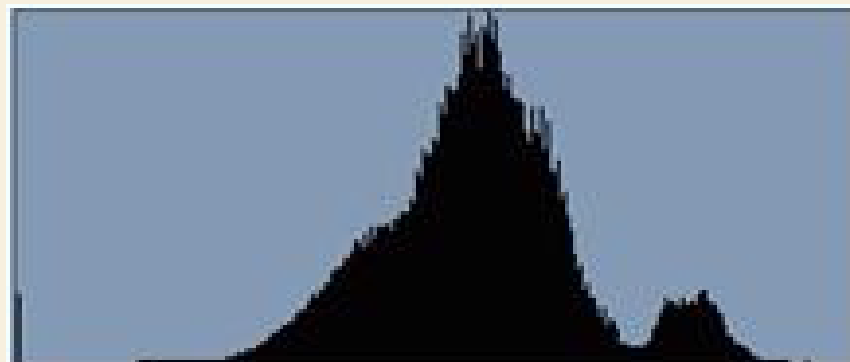
## ГИСТОГРАММЫ: ИДЕЯ

Компенсация узкого диапазона

яркостей – линейное  
растяжение:



# ГИСТОГРАММЫ: ИДЕЯ



```

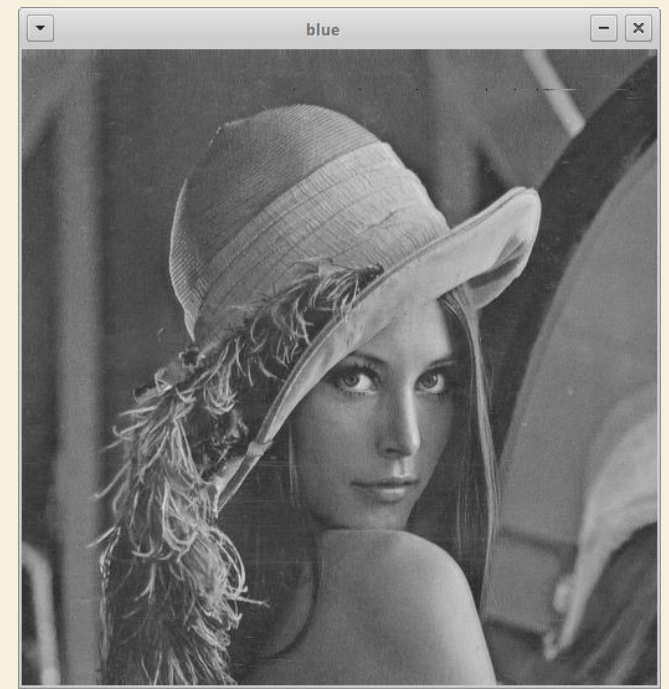
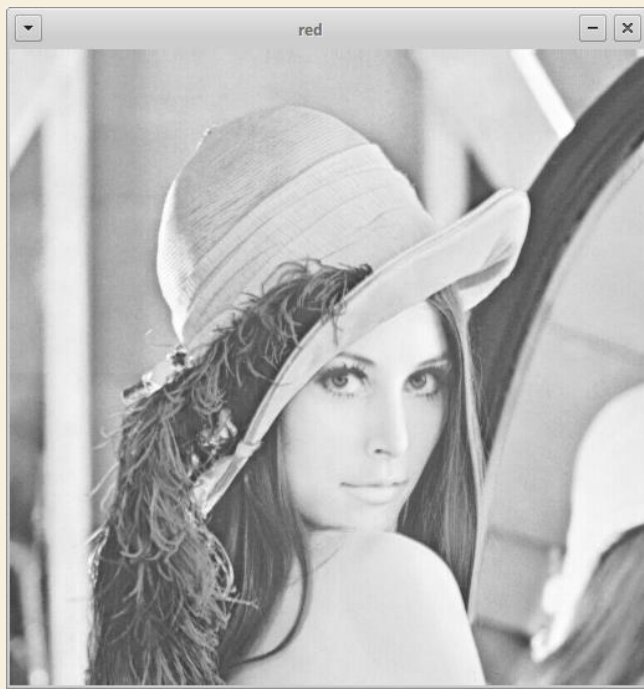
src = imread( imageName, IMREAD_COLOR ); // 3-канальное изображение
if( src.empty() )
{ return -1; }
vector<Mat> bgr_planes;
split( src, bgr_planes );
int histSize = 256;
float range[] = { 0, 256 } ;
const float* histRange = { range };
bool uniform = true; bool accumulate = false;
Mat b_hist, g_hist, r_hist;
calcHist( &bgr_planes[0], 1, 0, Mat(), b_hist, 1, &histSize, &histRange, uniform, accumulate );
calcHist( &bgr_planes[1], 1, 0, Mat(), g_hist, 1, &histSize, &histRange, uniform, accumulate );
calcHist( &bgr_planes[2], 1, 0, Mat(), r_hist, 1, &histSize, &histRange, uniform, accumulate );
int hist_w = 512; int hist_h = 400;
int bin_w = cvRound( (double) hist_w/histSize );
Mat histImage( hist_h, hist_w, CV_8UC3, Scalar( 0,0,0) );
normalize(b_hist, b_hist, 0, histImage.rows, NORM_MINMAX, -1, Mat() );
normalize(g_hist, g_hist, 0, histImage.rows, NORM_MINMAX, -1, Mat() );
normalize(r_hist, r_hist, 0, histImage.rows, NORM_MINMAX, -1, Mat() );
for( int i = 1; i < histSize; i++ )
{
    line( histImage, Point( bin_w*(i-1), hist_h - cvRound(b_hist.at<float>(i-1)) ),
          Point( bin_w*(i), hist_h - cvRound(b_hist.at<float>(i)) ),
          Scalar( 255, 0, 0), 2, 8, 0 );
    line( histImage, Point( bin_w*(i-1), hist_h - cvRound(g_hist.at<float>(i-1)) ),
          Point( bin_w*(i), hist_h - cvRound(g_hist.at<float>(i)) ),
          Scalar( 0, 255, 0), 2, 8, 0 );
    line( histImage, Point( bin_w*(i-1), hist_h - cvRound(r_hist.at<float>(i-1)) ),
          Point( bin_w*(i), hist_h - cvRound(r_hist.at<float>(i)) ),
          Scalar( 0, 0, 255), 2, 8, 0 );
}
namedWindow("calcHist Demo", WINDOW_AUTOSIZE );
imshow("calcHist Demo", histImage );

```

## ПОСТРОЕНИЕ ГИСТОГРАММ

Мы делаем три вещи:

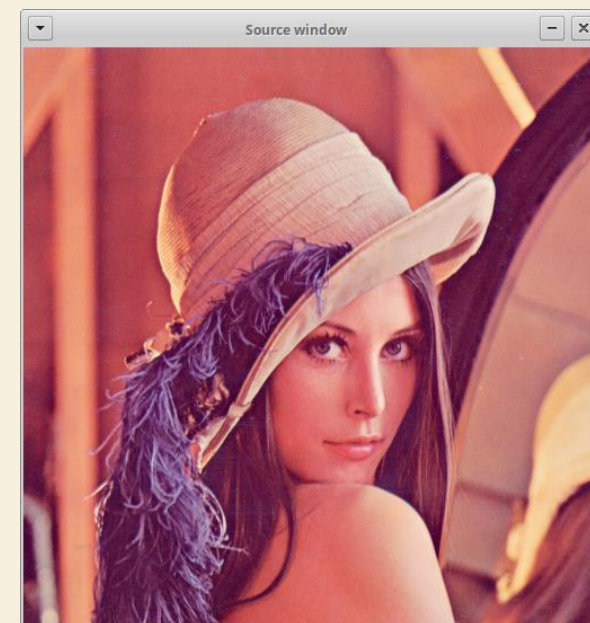
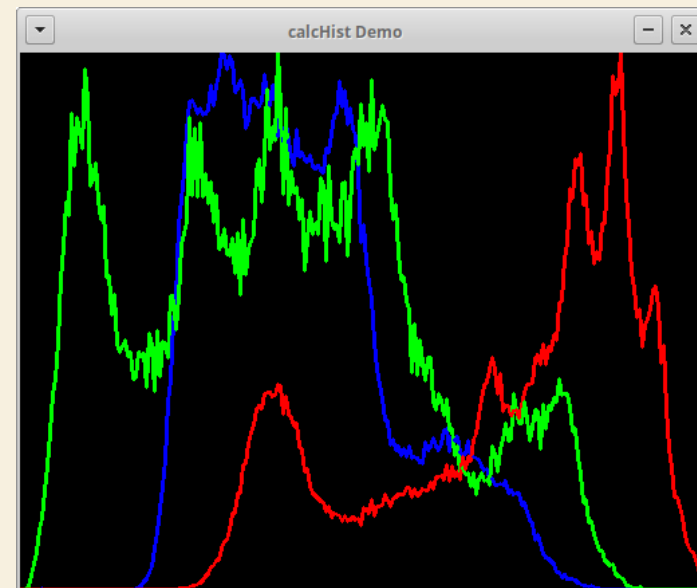
1. Вычисляем гистограмму
2. Нормализуем её
3. Рисуем отдельными линиями



# РАЗЛОЖЕНИЕ ПО КАНАЛАМ

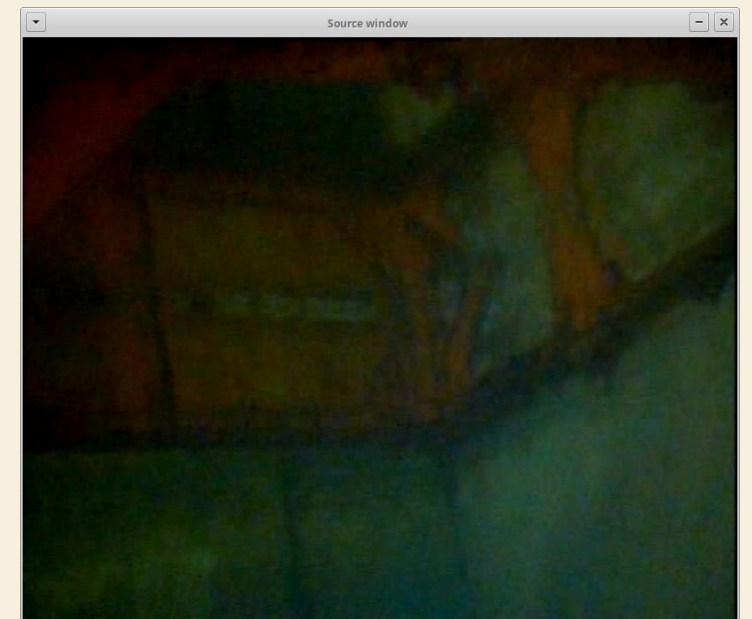
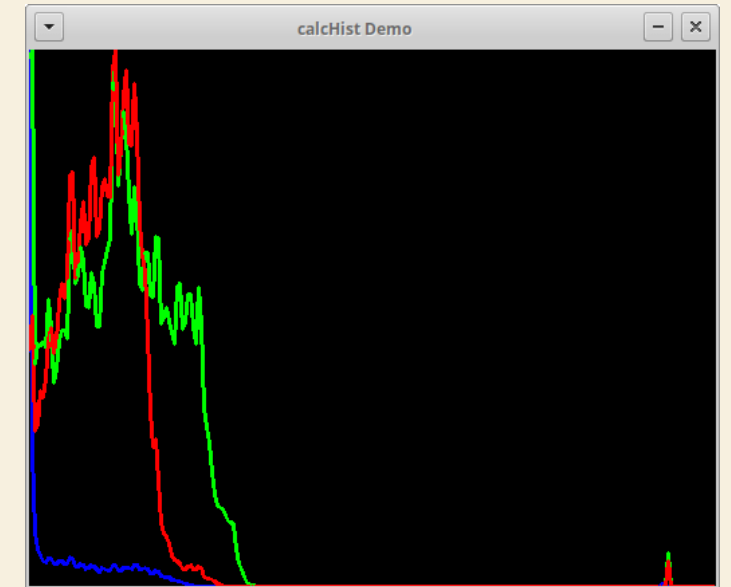
# РЕЗУЛЬТАТ

- Видно, что яркость в зелёном канале ниже, чем в остальных
- Яркость же в красном канале максимальна
- Это связано с красноватым оттенком большей части изображения
- На изображении вообще нет зелёных элементов, но зелёные компоненты есть в других цветах — например, в белом



# ФРАГМЕНТ ПОДВОДНОГО ИЗОБРАЖЕНИЯ

- В целом мы видим потёмки
- Изображение тёмное, поэтому большинство элементов на гистограмме собраны около нуля
- Возникает вопрос – как можно это поправить?





**ПОПРОБУЕМ УЛУЧШИТЬ  
ИЗОБРАЖЕНИЕ!**



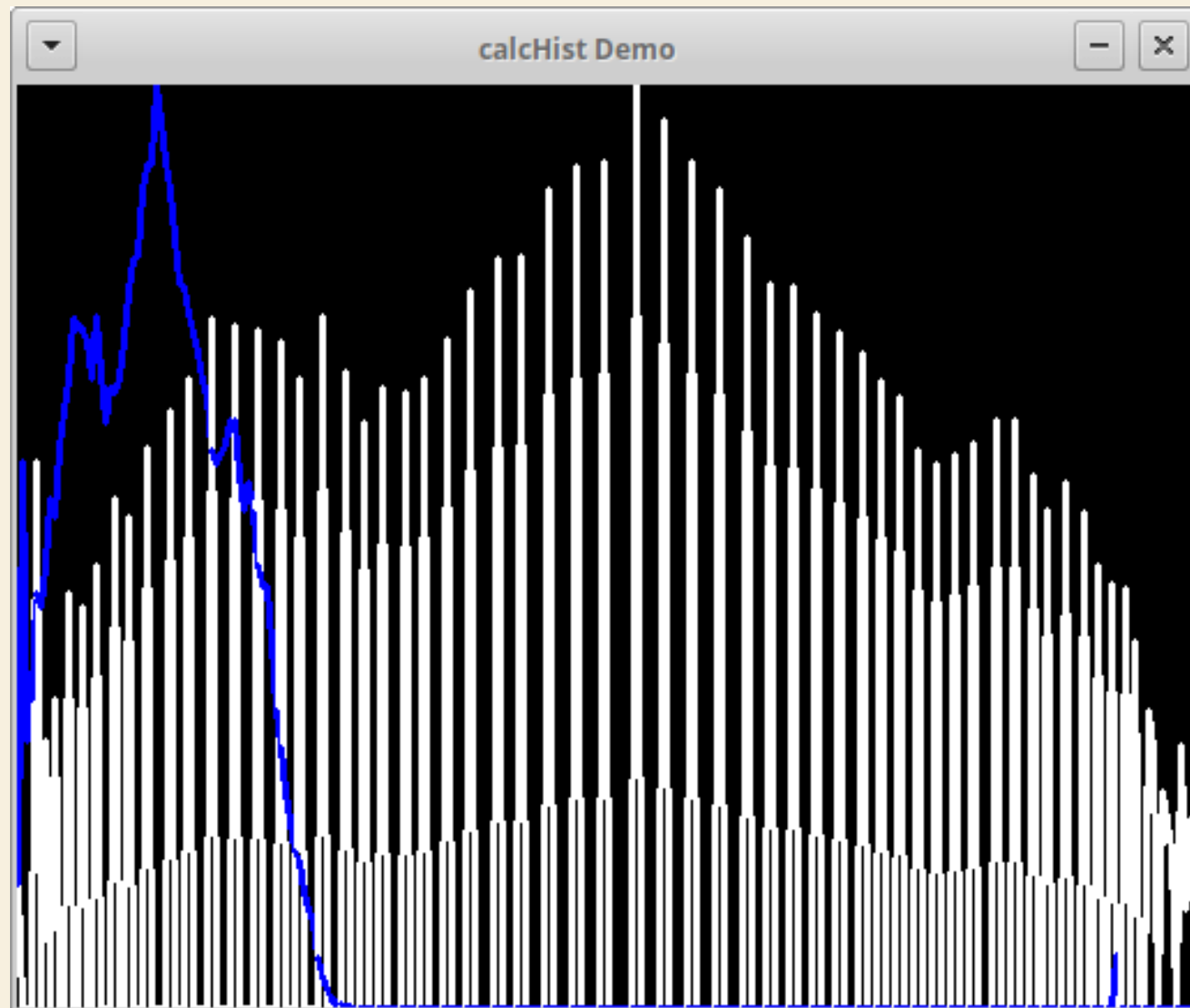
```
int main(int argc, char** argv)
{
    Mat src, dst;
    String imageName( "Picture1.png" );
    if (argc > 1)
    {
        imageName = argv[1];
    }
    src = imread( imageName, IMREAD_COLOR );
    if( src.empty() )
    { return -1; }
    Mat grayscale, grayEnh;
    cvtColor(src, grayscale, CV_BGR2GRAY);
    imshow("Source window", grayscale);
    equalizeHist(grayscale, grayEnh);
    imshow("Grayscale", grayEnh);
    waitKey(0); // ждём нажатия на любую кнопку
    return 0;
}
```

## «РАСТЯЖЕНИЕ» ГИСТОГРАММЫ

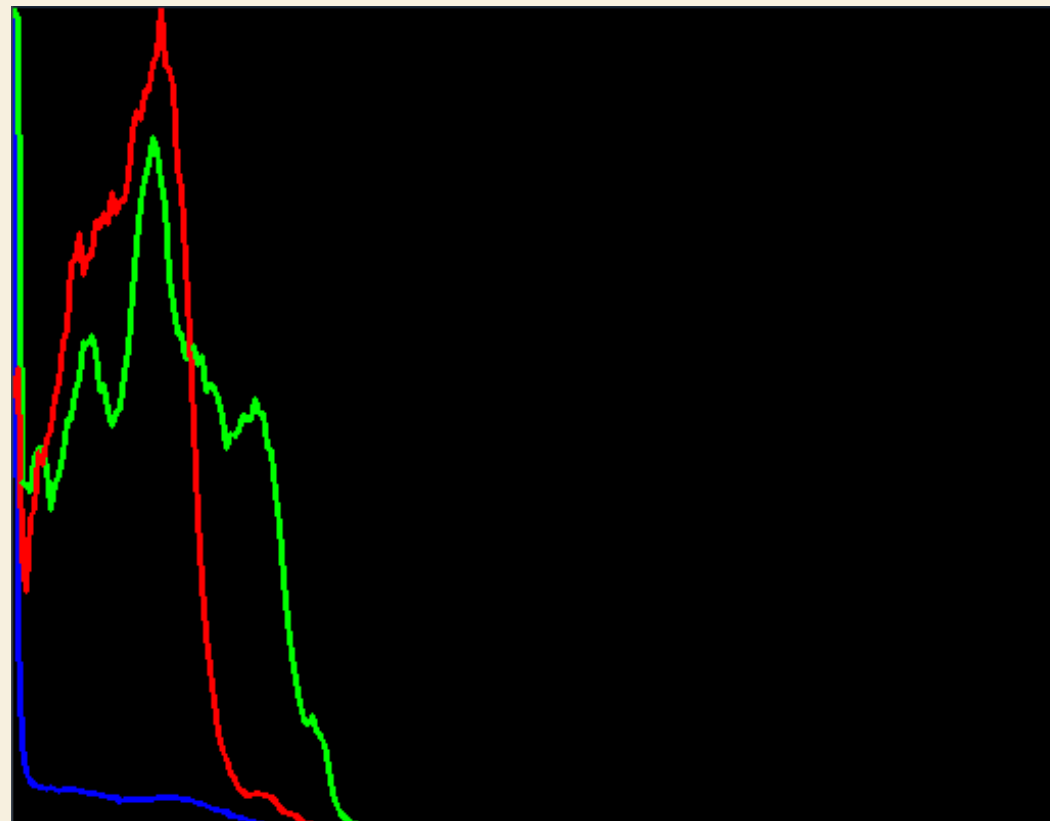
# ЧТО ПОЛУЧИЛОСЬ



# ГИСТОГРАММА

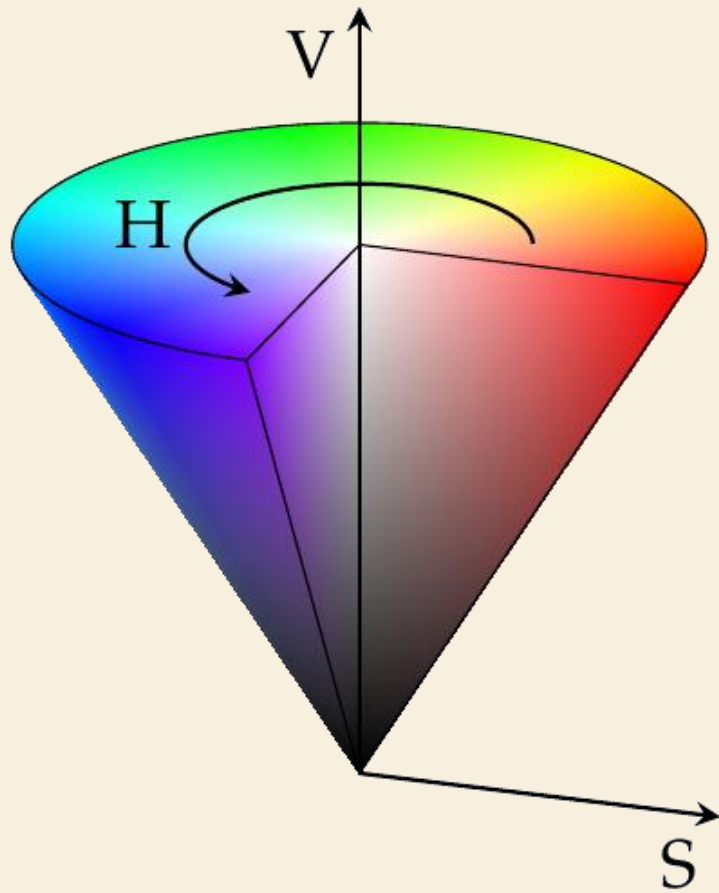


# ГИСТОГРАММЫ: А ЧТО ЖЕ С ЦВЕТОМ?





# ГИСТОГРАММЫ: HSV



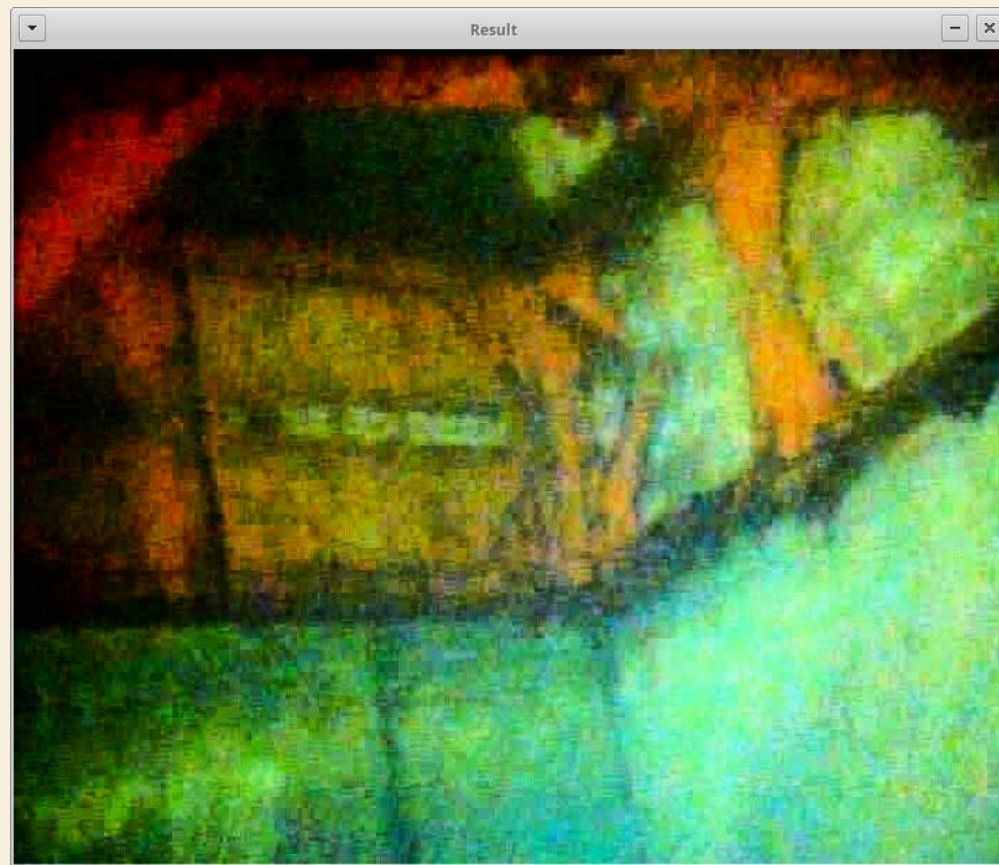
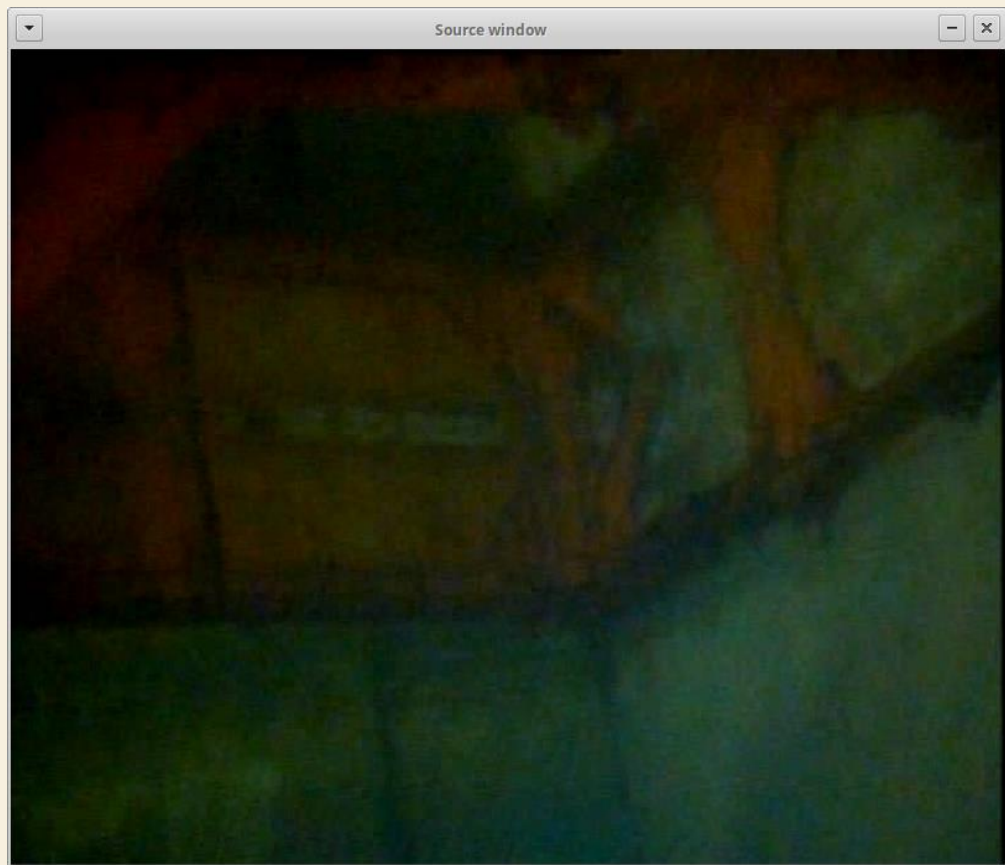
Hue (Тон), Saturation(Насыщенность), Value (Интенсивность)

Очевидно, что в данном случае, нас интересует Value!

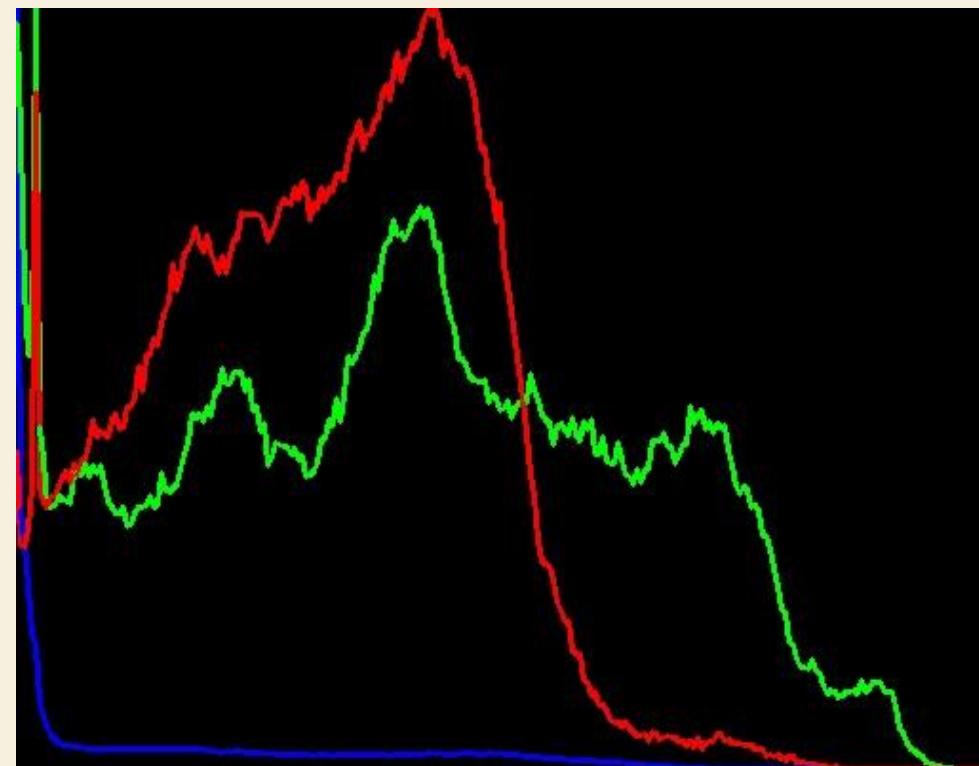
```
int main(int argc, char** argv)
{
    Mat src, dst;
    String imageName( "Picture1.png" ); // by default
    if (argc > 1)
    {
        imageName = argv[1];
    }
    src = imread( imageName, IMREAD_COLOR );
    if( src.empty() )
    { return -1; }
    Mat result;
    cvtColor(src, result, CV_BGR2HSV);
    Mat channels[3];
    split( result, channels);
    equalizeHist(channels[2],channels[2]);
    merge(channels,3, result);
    cvtColor(result,result,CV_HSV2BGR);
    imshow("Result", result);
}
```

«РАСТЯЖЕНИЕ»  
ОДНОГО КАНАЛА

# ЧТО ПОЛУЧИЛОСЬ



# ГИСТОГРАММЫ: А ЧТО ЖЕ С ЦВЕТОМ?



# НЕ ВСЕГДА ПОЛУЧАЕТСЯ...





# НЕЛИНЕЙНАЯ КОРРЕКЦИЯ

